Pending Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-24. (Cancelled).

Claim 25. (Previously Presented) A method for retiring instructions in a processor that executes a group of instructions, one or more of the group of instructions executed out of a program order, comprising:

assigning tags to instructions;

determining whether an executed instruction is retirable;

storing results of executed instructions in an index-addressable temporary buffer, wherein at least part of a tag assigned to an instruction indicates a location in said index-addressable temporary buffer where an execution result for the instruction is to be stored; and

retiring approximately simultaneously a group of retirable instructions, wherein said retiring comprises transferring execution results of said group of retirable instructions from said index-addressable temporary buffer to a register array, wherein said execution results of said group of retirable instructions are retrieved from said index-addressable temporary buffer based on at least part of each tag assigned to an instruction in said group of retirable instructions.

Claim 26. (Previously Presented) The method of claim 25, wherein said assigning comprises assigning a tag to an instruction wherein the value of a tag is based on a location of said instruction in an instruction window.

Claim 27. (Previously Presented) The method of claim 25, further comprising:

retiring instructions executed in program order by transferring an execution result directly from a functional unit to said register array.

Claim 28. (Previously Presented) The method of claim 25, wherein said retiring comprises retiring a group of instructions that includes at least two instructions.

Claim 29. (Previously Presented) The method of claim 25, wherein said retiring comprises retiring a group of instructions that includes four instructions.

Claim 30. (Previously Presented) The method of claim 25, wherein said retiring comprises simultaneously transferring execution results of more than one instruction from said index-addressable temporary buffer to said register array.

Claim 31. (Previously Presented) The method of claim 25, further comprising:

assigning done flags to the instruction or instructions determined to have been executed;

wherein said determining whether an executed instruction is retirable comprises checking said done flags to determine whether all instructions appearing earlier in the program order have completed.

Claim 32. (Previously Presented) The method of claim 25, wherein said retiring comprises simultaneously transferring execution results of more than one instruction from said index-addressable temporary buffer to said register array in a single clock cycle.

Claim 33. (Previously Presented) A method for retiring instructions in a processor that executes a group of instructions, one or more of the group of instructions executed out of a program order, comprising:

determining whether an executed instruction is retirable;

storing results of executed instructions in a temporary buffer, wherein at least one of said instructions is executed out of a program order;

transferring execution results of at least one instruction from said temporary buffer to a register array; and

transferring at least one execution result directly from a functional unit to said register array.

Claim 34. (Previously Presented) The method of claim 33, wherein said storing comprises storing results of executed instructions in an index-addressable temporary buffer, wherein execution results in said index-addressable temporary buffer are addressable by tags that include an address of a storage location in said index-addressable temporary buffer.

Claim 35. (Previously Presented) The method of claim 34, further comprising:

generating one or more tags to specify the storage location of execution results in said index-addressable temporary buffer, wherein the value of a tag for an instruction is based on a location of said instruction in an instruction window.

Claim 36. (Previously Presented) The method of claim 33, wherein said transferring comprises transferring execution results of more than one instruction approximately simultaneously from said temporary buffer to said register array.

Claim 37. (Previously Presented) The method of claim 33, wherein said transferring comprises transferring execution results of at least two instructions from said temporary buffer to said register array in a single clock cycle.

Claim 38. (Previously Presented) The method of claim 33, wherein said transferring comprises transferring execution results of four instructions from said temporary buffer to said register array in a single clock cycle.

Claim 39. (Previously Presented) The method of claim 33, further comprising:

assigning done flags to one or more instructions determined to have been executed;

wherein said determining whether an executed instruction is retirable comprises checking said done flags to determine whether all instructions appearing earlier in the program order have completed.

Claim 40. (Previously Presented) A superscalar processor configured to execute, out of a program order, a plurality of instructions from a group of instructions, the processor comprising:

a superscalar register renaming circuit configured to associate, in a superscalar manner, unique addresses with each instruction from the group of instructions;

a plurality of functional units configured to execute instructions from the group of instructions out of the program order;

a buffer configured to store execution results of instructions from the group of instructions, wherein the unique address associated with each instruction indicates a static location in the buffer where an execution result for each instruction is to be stored;

an array including a plurality of array locations referenced to provide execution results of instructions that have been retired;

a retirement control block configured to determine whether instructions from the group of instructions that are executed can be retired; and

a superscalar instruction retirement unit configured to retire, in a superscalar manner, a group of instructions that can be retired by associating a respective array location in the array to an execution result of each instruction in the group of instructions that can be retired, wherein the execution result of each instruction in the group of instructions that can be retired is stored in a respective static location in the buffer.

Claim 41. (Previously Presented) The superscalar processor of claim 40:

wherein the superscalar register renaming circuit is further configured to approximately simultaneously associate more than one of the respective unique addresses; and

wherein the superscalar instruction retirement unit is further configured to approximately simultaneously associate respective array locations in the array to the execution result of each instruction in the group of instructions that can be retired.

Claim 42. (Previously Presented) The superscalar processor of claim 40:

wherein the superscalar register renaming circuit is further configured to concurrently associate more than one of the respective unique addresses; and

wherein the superscalar instruction retirement unit is further configured to concurrently associate respective array locations in the array to the execution result of each instruction in the group of instructions that can be retired.

Claim 43. (Previously Presented) The superscalar processor of claim 40:

wherein the superscalar register renaming circuit is further configured to associate, substantially in parallel, more than one of the respective unique addresses; and

wherein the superscalar instruction retirement unit is further configured to associate, substantially in parallel, respective array locations in the array to the execution result of each instruction in the group of instructions that can be retired.

Claim 44. (Previously Presented) The superscalar processor of claim 40:

wherein each instruction from the group of instructions specifies a respective array location in the array; and

wherein the respective unique address for each instruction from the group of instructions is independent of the respective array location in the array specified in each instruction from the group of instructions.

Claim 45. (Previously Presented) The superscalar processor of claim 40:

wherein each instruction from the group of instructions specifies a respective array location in the array; and

wherein the respective unique address for each instruction from the group of instructions is associated without regard to the respective array location in the array specified in each instruction from the group of instructions.

Claim 46. (Previously Presented) The superscalar processor of claim 40:

wherein each instruction from the group of instructions specifies a respective array location in the array;

wherein the superscalar register renaming circuit is further configured to associate a first unique address with a first instruction specifying a first array location, and configured to associate the first unique address with a second instruction specifying a second array location; and

wherein the first array location and the second array location are different.

Claim 47. (Previously Presented) The superscalar processor of claim 40:

wherein each instruction from the group of instructions specifies a respective array location in the array;

wherein the superscalar register renaming circuit is further configured to associate a first and a second unique address respectively with a first instruction and a second instruction specifying respectively a first array location and a second array

location, and configured to associate the first unique address with a third instruction specifying a third array location; and

wherein the first array location and the third array location are different.

Claim 48. (Previously Presented) The superscalar processor of claim 47:

wherein the first instruction appears earlier in the program order than the second instruction;

wherein the second instruction is executed before the first instruction; and

wherein an execution result of the second instruction is stored in the buffer before an execution result of the first instruction is stored in the buffer.

Claim 49. (Previously Presented) The superscalar processor of claim 48:

wherein the buffer is an index addressable buffer;

wherein the buffer includes a first static location identified by the first unique address where the execution result of the first instruction is stored; and

wherein the buffer includes a second static location identified by the second unique address where the execution result of the second instruction is stored.

Claim 50. (Previously Presented) The superscalar processor of claim 49 wherein the first static location identified by the first unique address is where an execution result of the third instruction is stored.

Claim 51. (Previously Presented) The superscalar processor of claim 47 wherein the group of instructions that can be retired comprises at least two instructions.

Claim 52. (Previously Presented) The superscalar processor of claim 47:

wherein a first plurality of instructions includes the first instruction;

wherein a second plurality of instructions includes the second instruction; and

wherein the first plurality of instructions appears earlier in the program order relative to the second plurality of instructions.

Claim 53. (Previously Presented) The superscalar processor of claim 47:

wherein the associating the respective array location in the array to the execution result of each instruction in the group of instructions that can be retired comprises updating values stored in the respective array locations; and

wherein the array is referenced to provide an in order state of the superscalar processor.

Claim 54. (Previously Presented) A method for retiring instructions in a superscalar processor configured to execute a group of instructions out of a program order, the method comprising:

determining, in a superscalar manner, a first static location in a buffer where an execution result of a first instruction is to be stored and a second static location in the buffer where an execution result of a second instruction is to be stored, wherein the first instruction appears earlier in the program order than the second instruction;

storing the execution result of the second instruction in the buffer at the second static location;

storing the execution result of the first instruction in the buffer at the first static location wherein the second instruction is executed out of the program order with respect to the first instruction;

determining whether the first instruction can be retired;

determining whether the second instruction can be retired;

retiring, in a superscalar manner, the first instruction and the second instruction by associating locations in an array to the execution results of the first instruction stored in the buffer and the execution result of the second instruction stored in the buffer, wherein the execution result of the first instruction and the execution result of the second instruction are respectively identified within the buffer at the first static location and the second static location.

Claim 55. (Previously Presented) The method of claim 54:

wherein determining the first static location and the second static location comprises determining approximately simultaneously the first static location and the second static location; and

wherein retiring the first instruction and the second instruction comprises associating approximately simultaneously the locations in the array to the execution result of the first instruction stored in the buffer and the execution result of the second instruction stored in the buffer.

Claim 56. (Previously Presented) The method of claim 54:

wherein determining the first static location and the second static location comprises completing the determining of the first static location and the second static location in the same clock cycle; and

wherein retiring the first instruction and the second instruction comprises completing the associating of the locations in the array to the execution result of the first instruction stored in the buffer and the execution result of the second instruction stored in the buffer in the same clock cycle.

Claim 57. (Previously Presented) The method of claim 54:

wherein determining the first static location and the second static location comprises concurrently determining the first static location and the second static location; and

wherein retiring the first instruction and the second instruction comprises concurrently associating the locations in the array to the execution result of the first instruction stored in the buffer and the execution result of the second instruction stored in the buffer.

Claim 58. (Previously Presented) The method of claim 54:

wherein determining the first static location and the second static location comprises determining substantially in parallel the first static location and the second static location; and

wherein retiring the first instruction and the second instruction comprises associating substantially in parallel the locations in the array to the execution result of

the first instruction stored in the buffer and the execution result of the second instruction stored in the buffer.

Claim 59. (Previously Presented) The method of claim 54:

wherein the first instruction specifies a first location in the array;

wherein the second instruction specifies a second location in the array; and

wherein the superscalar determining of the first static location and the second static location is independent of the first location specified in the first instruction and the second location specified in the second instruction.

Claim 60. (Previously Presented) The method of claim 54:

wherein the first instruction specifies a first location in the array;

wherein the second instruction specifies a second location in the array; and

wherein the superscalar determining of the first static location and the second static location is without regard to the first location specified in the first instruction and the second location specified in the second instruction.

Claim 61. (Previously Presented) The method of claim 54 further comprising:

subsequent to determining of the first static location and the second static location, determining a third static location in the buffer where an execution result of a third instruction is to be stored;

wherein the first instruction, the second instruction, and the third instruction respectively specify a first location in the array, a second location in the array, and a third location in the array;

wherein the first static location and the third static location are identical locations, and

wherein the first location in the array and the third location in the array are different.

Claim 62. (Previously Presented) The method of claim 61 further comprising:

receiving a first plurality of instructions and a second plurality of instructions;

wherein the first plurality of instructions appears earlier in the program order than the second plurality of instructions;

wherein the first plurality of instructions comprises the first instruction; and

wherein the second plurality of instructions comprises the second instruction.

Claim 63. (Previously Presented) The method of claim 61 wherein determining whether the second instruction can be retired comprises determining that the execution result of the first instruction is stored in the buffer.

Claim 64. (Previously Presented) The method of claim 61 further comprising referring to the first location in the array to determine the execution result of the first instruction.

Claim 65. (Previously Presented) The method of claim 61 wherein the static locations in the buffer where the execution results of the first instruction, the second instruction, and the third instruction are to be stored comprise physical destinations.

Claim 66. (Previously Presented) The method of claim 65 wherein the buffer is index addressable.

Claim 67. (Previously Presented) The method of claim 61:

wherein associating of the locations in the array to the execution results comprises updating values stored in the locations in the array; and

wherein the array is referenced to provide an in order state of the superscalar processor.

Claim 68. (Previously Presented) A computer system including a memory configured to store instructions, the instructions having a program order, the system comprising:

a processor coupled to the memory, the processor comprising

a superscalar register renaming portion configured to determine, in a superscalar manner, an address associated with each instruction from a group of instructions;

a buffer coupled to the superscalar register renaming portion and configured to store execution results of instructions from the group of instructions at locations specified by the address associated with each instruction;

a plurality of functional units coupled to the buffer and configured to execute instructions from the group of instructions out of the program order;

an array including a plurality of locations configured to identify execution results of instructions that are retired;

a control block portion configured to determine whether instructions that are executed can be retired; and

a superscalar instruction retirement portion coupled to the control block portion and coupled to the array, the superscalar instruction retirement portion configured to retire, in a superscalar manner, a group of instructions that can be retired by associating a respective array location in the array to an execution result of each instruction in the group of instructions that can be retired, wherein the execution result of each instruction in the group of instructions that can be retired is stored at a respective specified location in the buffer.

Claim 69. (Previously Presented) The computer system of claim 68:

wherein the superscalar register renaming portion is further configured to approximately simultaneously determine more than one respective addresses; and

wherein the superscalar instruction retirement portion is further configured to approximately simultaneously associate respective array locations in the array to the execution result of each instruction in the group of instructions that can be retired.

Claim 70. (Previously Presented) The computer system of claim 68:

wherein the superscalar register renaming portion is further configured to concurrently determine more than one respective addresses; and

wherein the superscalar instruction retirement portion is further configured to concurrently associate respective array locations in the array to the execution result of each instruction in the group of instructions that can be retired.

Claim 71. (Previously Presented) The computer system of claim 68:

wherein the superscalar register renaming portion is further configured to determine, substantially in parallel, more than one respective addresses; and

wherein the superscalar instruction retirement portion is further configured to associate, substantially in parallel, respective array locations in the array to the execution result of each instruction in the group of instructions that can be retired.

Claim 72. (Previously Presented) The computer system of claim 68:

wherein each instruction from the group of instructions specifies a respective array location in the array; and

wherein a respective address for each instruction from the group of instructions is independent of a respective array location in the array specified in each instruction from the group of instructions.

Claim 73. (Previously Presented) The computer system of claim 68:

wherein each instruction from the group of instructions specifies a respective array location in the array; and

wherein a respective address for each instruction from the group of instructions is associated without regard to a respective array location in the array specified in each instruction from the group of instructions.

Claim 74. (Previously Presented) The computer system of claim 68:

wherein each instruction from the group of instructions specifies a respective array location in the array;

wherein the superscalar register renaming portion is further configured to associate a first address with a first instruction specifying a first array location, and configured to associate the first address with a second instruction specifying a second array location; and

wherein the first array location and the second array location are different.

Claim 75. (Previously Presented) The computer system of claim 68:

wherein each instruction from the group of instructions specifies a respective array location in the array;

wherein the superscalar register renaming portion is configured to associate a first and a second address respectively with a first instruction and a second instruction specifying respectively a first array location and a second array location, and configured to associate the first address with a third instruction specifying a third array location; and

wherein the first array location and the third array location are different.

Claim 76. (Previously Presented) The computer system of claim 75:

wherein the first instruction appears earlier in the program order than the second instruction;

wherein the second instruction is executed before the first instruction; and

wherein an execution result of the second instruction is stored in the buffer before an execution result of the first instruction is stored in the buffer.

Claim 77. (Previously Presented) The computer system of claim 75:

wherein the first address identifies a first location in the buffer where an execution result of the first instruction is to be stored; and

wherein the second address identifies a second location in the buffer where an execution result of the second instruction is to be stored.

Claim 78. (Previously Presented) The computer system of claim 77 wherein the first address identifies where an execution result of the third instruction is to be stored.

Claim 79. (Previously Presented) The computer system of claim 77 wherein the first location and the second location are static within the buffer.

Claim 80. (Previously Presented) The computer system of claim 77 wherein the second instruction cannot be retired until the execution result of the first instruction is stored at the first location in the buffer.

Claim 81. (Previously Presented) The computer system of claim 75:

wherein a first plurality of instructions includes the first instruction;

wherein a second plurality of instructions includes the second instruction; and

wherein the first plurality of instructions appears earlier in the program order than the second plurality of instructions.

Claim 82. (Previously Presented) The computer system of claim 75:

wherein associating respective array location in the array to the execution result of each instruction in the group of instructions that can be retired comprises updating values stored in respective array locations; and

wherein the array is referenced to provide an in order state of the processor.